

(19)



JAPANESE PATENT OFFICE

PATENT ABSTRACTS OF JAPAN

(11) Publication number: 2001084137 A

(43) Date of publication of application: 30.03.01

(51) Int. Cl

G06F 9/06

(21) Application number: 11260766

(71) Applicant: MATSUSHITA ELECTRIC IND CO LTD

(22) Date of filing: 14.09.99

(72) Inventor: KURAUCHI NOBUKAZU  
UEISHI NARITOSHI  
SUZUKI MOTOFUMI

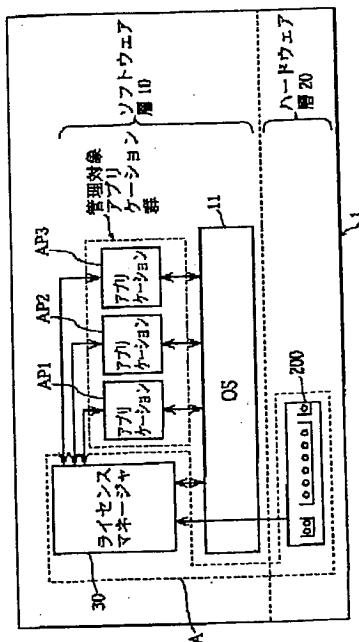
(54) DEVICE AND METHOD FOR MANAGING  
LICENSE

(57) Abstract:

PROBLEM TO BE SOLVED: To provide a license managing device for preventing license violation without causing the size increase of an application or the performance deterioration of the application and a system.

SOLUTION: In a license managing device A, a license manager 30 operating as a process different from an application to be managed operates license check for confirming the presence or absence of a license key 200 and the presence or absence of any illegal use at least once for a prescribed time based on held license information and the operating situation of the application to be managed. The license check processing is operated in a batch by the license manager regardless of the number of the applications to be managed.

COPYRIGHT: (C)2001,JPO



(19) 日本国特許庁 (JP)

## (12) 公開特許公報 (A)

(11) 特許出願公開番号

特開2001-84137

(P2001-84137A)

(43) 公開日 平成13年3月30日 (2001.3.30)

(51) Int.Cl.<sup>7</sup>

G 06 F 9/06

識別記号

550

F I

G 06 F 9/06

テマコト<sup>\*</sup> (参考)

550Z 5B076

審査請求 未請求 請求項の数15 O L (全 22 頁)

(21) 出願番号 特願平11-260766

(22) 出願日 平成11年9月14日 (1999.9.14)

(71) 出願人 000005821

松下電器産業株式会社

大阪府門真市大字門真1006番地

(72) 発明者 倉内 伸和

愛知県名古屋市中区栄2丁目6番1号 白川ビル別館5階 株式会社松下電器情報システム名古屋研究所内

(72) 発明者 上石 成銳

愛知県名古屋市中区栄2丁目6番1号 白川ビル別館5階 株式会社松下電器情報システム名古屋研究所内

(74) 代理人 100090446

弁理士 中島 司朗 (外1名)

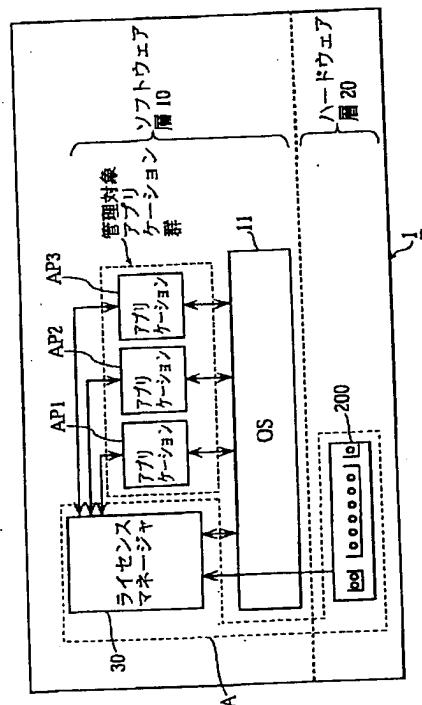
最終頁に続く

(54) 【発明の名称】 ライセンス管理装置およびライセンス管理方法

(57) 【要約】

【課題】 アプリケーションの肥大化、またアプリケーションおよびシステムのパフォーマンス低下を招くことなしに、ライセンス違反を防止できるライセンス管理装置を提供する。

【解決手段】 管理対象アプリケーションとは別個のプロセスとして動作するライセンスマネージャ30が、ライセンスキー200の有無およびこれが保持するライセンス情報と、管理対象アプリケーションの稼動状況とともに、少なくとも所定時間に1度は不正使用の有無を確認するライセンスチェックを行う。ライセンスチェック処理は、管理対象アプリケーションの数に関係なく、ライセンスマネージャがまとめて行う。



## 【特許請求の範囲】

【請求項1】 ライセンス管理の対象となる1以上の管理対象アプリケーションプログラムについてコンピュータシステム上での使用状況を管理するライセンス管理装置であつて、

前記1以上の管理対象アプリケーションプログラムについて、前記コンピュータシステムでの使用の可否を示すライセンス情報を保持する、ハードウェアのライセンスキーと、

前記コンピュータシステム上で、前記1以上の管理対象アプリケーションプログラムとは別のプロセスとして動作し、前記ライセンス情報をもとにライセンスチェックを行つて前記1以上の管理対象アプリケーションプログラムの不正使用を防止するマネージャ手段とを備えるライセンス管理装置。

【請求項2】 前記マネージャ手段は、前記ライセンスキーから前記ライセンス情報を読み出す読み出し手段と、

前記1以上の管理対象アプリケーションプログラムの前記コンピュータシステム上での実行状況を示す稼動情報取得手段と、

前記読み出し手段が読み出したライセンス情報と前記取得手段が取得した稼動情報とをもとに、前記1以上の管理対象アプリケーションプログラムの中に、前記コンピュータシステムでの動作が認可されていないにもかかわらず動作しているものがあれば、不正使用ありと判定する判定手段と、

前記判定手段が不正使用ありと判定した場合、稼働中である前記1以上の管理対象アプリケーションプログラムのうち、少なくとも前記判定手段によって前記コンピュータシステムでの動作が認可されていないにもかかわらず動作していると判定されたものを停止させるための処理を行う停止手段とを備えることを特徴とする請求項1に記載のライセンス管理装置。

【請求項3】 前記読み取り手段は、ライセンスキーがシステムに存在しなければ、その旨を前記判定手段に通知し、

前記判定手段は、ライセンスキーがシステムに存在しない場合、前記管理対象アプリケーションが1つ以上でも前記コンピュータシステムで稼動していれば不正使用ありと判定することを特徴とする請求項2に記載のライセンス管理装置。

【請求項4】 前記マネージャ手段によるライセンスチェックの実行タイミングを制御する制御手段を更に備えることを特徴とする請求項1乃至3のいずれかに記載のライセンス管理装置。

【請求項5】 前記コンピュータシステムにおける、前記1以上の管理対象アプリケーションプログラム以外のプロセスの稼動状況に関する情報を取得するプロセス情報取得手段を更に備え、

前記制御手段は、前記稼動情報取得手段が取得した情報に基づいて、前記マネージャ手段によるライセンスチェックの実行タイミングを変更することを特徴とする請求項4に記載のライセンス管理装置。

【請求項6】 前記1以上の管理対象アプリケーションプログラムはコールバック機能を有し、前記停止手段は、前記1以上の管理対象アプリケーションプログラムを停止させるための処理を行う場合に、前記1以上の管理対象アプリケーションプログラムに対して前記コールバック機能の実行を認めることを特徴とする請求項4または5に記載のライセンス管理装置。

【請求項7】 前記ライセンスキーの前記コンピュータシステムからの取り外しを監視し、取り外しを検知した場合は、その旨を前記制御手段に通知するキー監視手段を更に備え、

前記制御手段は、前記キー監視手段からの通知を受けると、前記マネージャ手段にライセンスチェック判定処理を実行させることを特徴とする請求項4乃至6のいずれかに記載のライセンス管理装置。

【請求項8】 前記1以上の管理対象アプリケーションプログラムのうち少なくとも1つは制限付き使用が可能であり、

前記停止手段は、前記検証手段が制限付き使用の可能な管理対象アプリケーションプログラムについて不正使用と判定した場合、停止処理を行わず、当該制限付き使用が可能な管理対象アプリケーションプログラムについて機能を一部制限することを特徴とする請求項4乃至7のいずれかに記載のライセンス管理装置。

【請求項9】 前記取得手段は、前記1以上の管理対象アプリケーションプログラムとの通信によって前記稼動情報を取得し、

前記稼動情報は、前記取得手段と前記1以上の管理対象アプリケーションプログラムとのプロセス間通信に用いられる領域であるポートの識別情報を含むことを特徴とする請求項4乃至8のいずれかに記載のライセンス管理装置。

【請求項10】 前記停止手段は同報通信機能を有し、前記ポートの識別情報を用いて前記1以上の管理対象アプリケーションプログラムに対して終了命令を同報送信することで、前記1以上の管理対象アプリケーションプログラムを停止させることを特徴とする請求項4乃至9のいずれかに記載のライセンス管理装置。

【請求項11】 前記コンピュータシステムは複数の端末がネットワークで接続されて成り、

前記ライセンスキーは、前記複数の端末のうち少なくとも1つに接続され、

前記1以上の管理対象アプリケーションプログラムは、前記複数の端末のうち少なくとも1つの上で実行されることを特徴とする請求項4乃至10のいずれかに記載のライセンス管理装置。

【請求項12】 前記停止手段はシステム権限を有し、前記コンピュータシステム権限を用いて、前記1以上の管理対象アプリケーションプログラムのプロセスを停止させることを特徴とする請求項4乃至11のいずれかに記載のライセンス管理装置。

【請求項13】 前記1以上の管理対象アプリケーションプログラムを子プロセスとして起動および停止させる子プロセス管理手段を更に有し、

前記取得手段は、前記子プロセス管理手段から前記稼動情報を取得し、

前記停止手段は、前記子プロセス管理手段に指示して前記1以上の管理対象アプリケーションプログラムを停止させることを特徴とする請求項4乃至8のいずれかに記載のライセンス管理装置。

【請求項14】 ライセンス管理の対象となる1以上の管理対象アプリケーションプログラムについてコンピュータシステム上での実行状況を管理するライセンス管理办法であって、

前記1以上の管理対象アプリケーションプログラムについて、前記コンピュータシステムでの使用の可否を示すライセンス情報を保持する、ハードウェアのライセンスキーから前記ライセンス情報を読み出す読み出しステップと、

前記1以上の管理対象アプリケーションプログラムの前記コンピュータシステム上での実行状況を示す稼動情報を取得する取得ステップと、

前記読み出しステップにおいて読み出されたライセンス情報と前記取得ステップにおいて取得された稼動情報をもとに、前記1以上の管理対象アプリケーションプログラムの中に、前記コンピュータシステムでの動作が認められるものがあるかないにもかかわらず動作しているものがあれば、不正使用ありと判定する判定ステップと、

前記判定ステップにおいてが不正使用ありと判定された場合、稼動中である前記1以上の管理対象アプリケーションプログラムのうち、少なくとも前記判定ステップにおいて前記コンピュータシステムでの動作が認められないものがあるかないにもかかわらず動作していると判定されたものを停止させるための処理を行う停止ステップとから成るライセンス管理办法。

【請求項15】 ライセンス管理の対象となる1以上の管理対象アプリケーションプログラムについてコンピュータシステム上での実行状況を管理するライセンス管理办法のプログラムを記録したコンピュータ読み取り可能な記録媒体であって、

前記プログラムは、

前記1以上の管理対象アプリケーションプログラムについて、前記コンピュータシステムでの使用の可否を示すライセンス情報を保持する、ハードウェアのライセンスキーから前記ライセンス情報を読み出す読み出しステップと、

前記1以上の管理対象アプリケーションプログラムの前記コンピュータシステム上での実行状況を示す稼動情報を取得する取得ステップと、

前記読み出しステップにおいて読み出されたライセンス情報と前記取得ステップにおいて取得された稼動情報をもとに、前記1以上の管理対象アプリケーションプログラムの中に、前記コンピュータシステムでの動作が認められないものがあるかないにもかかわらず動作しているものがあれば、不正使用ありと判定する判定ステップと、

前記判定ステップにおいてが不正使用ありと判定された場合、稼動中である前記1以上の管理対象アプリケーションプログラムのうち、少なくとも前記判定ステップにおいて前記コンピュータシステムでの動作が認められないものがあるかないにもかかわらず動作していると判定されたものを停止させるための処理を行う停止ステップとを含むことを特徴とする記録媒体。

#### 【発明の詳細な説明】

##### 【0001】

【発明が属する技術分野】 本発明は、ハードウェアのライセンスキーを用いて、アプリケーションのソフトウェアのライセンス管理を行うライセンス管理装置およびライセンス管理方法に関する。

##### 【0002】

【従来の技術】 従来、ソフトウェアについては、複製が容易であるという性質上、不正使用が行われやすい。そこで、配布元は、ソフトウェアの配布（販売、レンタル）にあたってはユーザーとの間にライセンス契約を結び、配布後もライセンス管理によって不正使用を防止する対策をとっている。アプリケーションソフトウェアのライセンス管理の一例として、主に企業などに大口ユーザーに対して行われるものに、ライセンスキーという専用のハードウェアを用いるものがある。アプリケーション使用者数はライセンス契約で定められ、配布元は使用者数分のライセンスキーをユーザーに配布する。

【0003】 ライセンスキーは、パーソナルコンピュータ（以下「PC」）のコネクタ（シリアル、パラレルなど）に接続して用いられるものであり、ライセンスキーが装着されたPC上でのライセンス管理の対象となるアプリケーションソフトウェア（以下「管理対象アプリケーション」という）の使用（インストールまたは稼動）の可否を示す情報を保持している。

【0004】 管理対象アプリケーションがライセンス契約に違反した形で使用されていないかどうかのチェック（以下「ライセンスチェック」という）が行われるタイミングとしては、アプリケーションのインストール時、アプリケーションの起動時、アプリケーションの稼働中、がある。図18は、インストール時のライセンスチェックの概要を示す。これはライセンスキーが装着されていないPCには管理対象アプリケーションをインストールできないようにするものである。ライセンスキーは

当該管理対象アプリケーションのインストールを認める情報を保持している。

【0005】ライセンスを受けたユーザは、アプリケーション配布媒体であるCD-ROM1810をPC1800のドライブ1820にセットし、キーボード1830から、アプリケーションに与えられた製品IDを入力して管理対象アプリケーションのハードディスクドライブ1860へのインストールを行う。この時、インストーラ1840（管理対象アプリケーションとともにCD-ROM1810に格納されている）は、ライセンスキー1850の有無をチェックし、ライセンスキー1850が装着されている場合にだけ、管理対象アプリケーションのインストールを実行する。

【0006】次いで、図19は、起動時に行われるライセンスチェックの概要を示す。これはライセンスキーが装着されていないPCで管理対象アプリケーションは起動できないようにするものである。ライセンスキーは当該管理対象アプリケーションの起動を認める情報を保持しているこの場合、ユーザがPC1900のキーボード1920を操作して管理対象アプリケーション1910の起動を指示すると、OS1940上でのプロセスとして管理対象アプリケーション1910の起動処理が開始される。この時、管理対象アプリケーション1910は、起動処理の一部としてライセンスキー1930の有無をチェックし、ライセンスキー1930が存在すれば、そのまま起動処理を継続するが、ライセンスキー1930が存在しなければ起動処理を停止する。

【0007】上記2種類のライセンスチェックは、組み合わせて実施されることが多い。また、いずれの場合も、1つのライセンスキーで1種類の管理対象アプリケーションを管理する場合について説明したが、複数の管理対象アプリケーションを1つのライセンスキーで管理することもある。その場合、ライセンスキーは管理対象アプリケーションごとに使用の可否を示す情報を保持しており、ライセンスチェックはライセンスキーの有無だけでなく、ライセンスキーが保持する情報の内容まで参照して行われる。

【0008】図18に示すインストール時のライセンスチェック、また図19に示す起動時のライセンスチェックについては、いったんインストールまたは起動してしまうと、その後はライセンスチェックが行われないため、不正使用される可能性が小さくない。例えば、インストール時または起動時のライセンスキーをPCに装着し、インストールまたは起動が完了した時点でライセンスキーをPCから取り外して他のPCに使いまわす、といった形の不正使用には対応できない。

【0009】そこで、管理対象アプリケーション稼働中（起動後）にライセンスチェックを実行する方法が必要となる。これは、管理対象アプリケーション内にライセンスチェック用の命令を埋め込み、管理対象アプリケ

ション実行中に適当な時間間隔をおいて、このライセンスチェック用の命令を実行するというものである。図20は、管理対象アプリケーション2000に埋め込まれたライセンスチェック用命令（以下「トリガ命令」という）2010を示すイメージ図である。トリガ命令2010の埋め込み箇所は、あまり頻繁でなく、しかも、ある程度周期的に実行されるような部分（例えば文書作成プログラムにおける保存処理のルーチン）となっている。また、プログラムの改ざんによる不正使用防止のために、トリガ命令は複数箇所に埋め込まれている。

【0010】図20に示すトリガ命令を用いたライセンスチェックでは、アプリケーション稼働中に、トリガ部分を実行するたびにライセンスチェックが行われるので、アプリケーション起動後にライセンスキーを取り外して他のPCで使いまわすといった不正使用は防止できる。

#### 【0011】

【発明が解決しようとする課題】しかしながら、図20に示すライセンスチェックの場合、そもそもトリガ命令は、アプリケーション本来の用途には無関係なものであるが、その無関係なトリガ命令のためにプログラムが肥大してしまう。さらに、多数の管理対象アプリケーションが同時に稼働している環境では、複数のアプリケーションのライセンスチェック（ライセンスキーへのアクセス）が同時に実行されると、バス上のトラフィックが増大してシステム全体のパフォーマンスが低下してしまう場合がある。また、稼働中の管理対象アプリケーションの数が増えるほどライセンスチェック処理が頻繁に実行されることになり、ライセンスチェック処理に占有される資源（CPUなど）が多くなる。すると、管理対象アプリケーション本来の処理（文書作成や表計算など）に用いられる資源が減り、アプリケーションのレベルでもパフォーマンスが低下する。

【0012】図21は、5種類の管理対象アプリケーションが1つのPC上で同時に稼働している状態を示す。各アプリケーションには、ライセンスチェック用のトリガ命令が埋め込まれている。PC2100上では、5種類の管理対象アプリケーションAP1～AP5がインストールされており、全てが稼働中である。PC2100のシリアルポート（図示せず）には、ライセンスキー2110が装着されており、ライセンスチェック時（トリガ命令2010実行時）、各アプリケーションAP1～AP5は、バス2120を介してライセンスキー2110へアクセスする。図21では、アプリケーションAP1～AP5がすべてライセンスチェックのためにライセンスキー2110へのアクセスを行い、バス2120上のトラフィックが増大している。バス2120はシステム上の他のプロセスも使用するので、このトラフィックの増大により、システム全体のパフォーマンスが低下する。

【0013】本発明は上記課題に鑑み、アプリケーションの肥大化、またアプリケーションおよびシステムのパフォーマンス低下を招くことなしに、ライセンス違反の不正使用を防止することのできるライセンス管理装置およびライセンス管理方法を提供することを目的とする。

#### 【0014】

【課題を解決するための手段】上記の目的を達成するために、本発明のライセンス管理装置は、ライセンス管理の対象となる1以上の管理対象アプリケーションプログラムについてコンピュータシステム上の使用状況を管理するライセンス管理装置であって、前記1以上の管理対象アプリケーションプログラムについて、前記コンピュータシステムでの使用の可否を示すライセンス情報を保持する、ハードウェアのライセンスキーと、前記コンピュータシステム上で、前記1以上の管理対象アプリケーションプログラムとは別のプロセスとして動作し、前記ライセンス情報をもとにライセンスチェックを行って前記1以上の管理対象アプリケーションプログラムの不正使用を防止するマネージャ手段とを備えることを特徴とする。

【0015】また、上記の目的を達成するために、本発明のライセンス管理方法は、ライセンス管理の対象となる1以上の管理対象アプリケーションプログラムについてコンピュータシステム上の実行状況を管理するライセンス管理方法であって、前記1以上の管理対象アプリケーションプログラムについて、前記コンピュータシステムでの使用の可否を示すライセンス情報を保持する、ハードウェアのライセンスキーから前記ライセンス情報を読み出す読み出しステップと、前記1以上の管理対象アプリケーションプログラムの前記コンピュータシステム上の実行状況を示す稼動情報を取得する取得ステップと、前記読み出しステップにおいて読み出されたライセンス情報と前記取得ステップにおいて取得された稼動情報をもとに、前記1以上の管理対象アプリケーションプログラムの中に、前記コンピュータシステムでの動作が認可されていないにもかかわらず動作しているものがあれば、不正使用ありと判定する判定ステップと、前記判定ステップにおいてが不正使用ありと判定された場合、稼働中である前記1以上の管理対象アプリケーションプログラムのうち、少なくとも前記判定ステップにおいて前記コンピュータシステムでの動作が認可されていないにもかかわらず動作していると判定されたものを停止させるための処理を行う停止ステップとを有することを特徴とする。

#### 【0016】

【発明の実施の形態】以下、本発明のライセンス管理装置に関する実施の形態を、図面を参照しながら説明する。なお、本実施の形態のライセンス管理装置によって監視されるライセンス契約の概要は以下の通りである。このライセンス契約は、企業ユーザに対して3種類のビ

ジネス用ソフトウェア（ワードプロセッサ、表計算、データベース管理）をまとめたソフトウェアパッケージがライセンス管理の対象となったもので、当該企業では10台のPC上でこのソフトウェアパッケージ使用できる。ただし、10台全てで3種類全てのソフトウェアを使用できる訳ではない。ワードプロセッサは10台全てで使用できるが、表計算は8台まで、データベース管理は3台までしか使用できない。そのため、10個のライセンスキーは、ライセンス内容によって3種類に分かれている。

(1) 3種類のアプリケーションすべての使用を認めるもの：3個

(2) ワードプロセッサと表計算の使用を認めるもの：5個

(3) ワードプロセッサのみ使用を認めるもの：2個  
ただし、インストールについては何台のPCにインストールしてもかまわない。ライセンスキーによって規制されるのは、管理対象アプリケーションを同時に稼動できるPCの数である。

【0017】なお、本実施の形態において想定している主なライセンス違反（不正使用）のパターンは以下の通りである。

(1) ライセンスキーなしでの使用：管理対象アプリケーション起動後にライセンスキーを取り外し、他のPCでの管理対象アプリケーションの起動に用いる。これを繰り返せば、10台を越えるPCで管理対象アプリケーションを稼動させることができる。

(2) ライセンスキーの交換：3種類のアプリケーションすべての使用を認めるライセンスキーで3種類の管理対象アプリケーションを起動させた後、ワードプロセッサのみ使用を認めるライセンスキーと取り替え、他のPCでさらに3種類の管理対象アプリケーションを起動させる。これを繰り返せば、10台のPCすべてで3種類の管理対象アプリケーションを起動させることができる。

【0018】(構成) 図1は、本実施の形態のライセンス管理装置が1台のPC上でどのように実現されるかを、上記ライセンス契約を結んだ企業ユーザにおける1ユーザ（1台のPC）を例にとって示す模式図である。本PCには、3種類のソフトウェア全てがインストールされており、ワードプロセッサと表計算の使用を認めるライセンスキーが装着されるものとする。

【0019】PC10は、ソフトウェア層11とハードウェア層12とを有し、ライセンス管理装置Aは、ハードウェア層12においてコネクタ（図示せず）の1つに装着されたライセンスキー21と、ソフトウェア層11においてOS110上で動作するライセンスマネージャ22とから成る。さらにOS110上では、ライセンス管理の対象となる3種類のアプリケーションAP1、AP2、AP3が動作する。なお、図示はしないが、OS

110上では、ライセンス管理の対象とならないアプリケーションなども動作する。

【0020】ライセンスマネージャ30は、アプリケーションAP1、AP2、AP3と同じ媒体(CD-ROM)で提供されたもので、アプリケーションAP1、AP2、AP3のPC10へのインストール時に、あわせてインストールされたものである。ライセンスキーコードは、PC10のコネクタ(シリアル、パラレル、USB: Universal Serial Busなど)に装着されるハードウェアであり、ここではシリアルコネクタ(図示せず)に装着されている。

【0021】ライセンスキーコードは、内蔵メモリ内に、ライセンスチェックに用いられるライセンス情報(当該ライセンスキーコードが装着されたPCでは、どの管理対象アプリケーションが使用できるかを示す情報)を保持とともに、外からの要求に応じて、この情報を秘密のプロトコルに従って出力する回路を有している。図2は、ライセンス情報の構成を示す。

【0022】ライセンス情報210は、ライセンスキーアイデンティフィケーション番号211とライセンスデータ212から成る。ライセンスキーアイデンティフィケーション番号211は、個々のライセンスキーコードに付加されたユニークなシリアル番号である。このシリアル番号は、製造元や規格の情報を含み、ライセンスマネージャ30は、シリアル番号の有無とその内容どちら、当該キーが自分とともにライセンス管理装置を構成する自社のライセンスキーコードであること(ライセンスキーコードの正当性)を認識する。

【0023】ライセンスデータ212は、当該ライセンスキーコードが装着されたPCでの使用が認められている管理対象アプリケーションのアプリケーションIDである。アプリケーションIDは、製造元が管理対象アプリケーションに付加した識別情報(製品番号およびバージョン番号)である。ここでライセンスデータ212は2件であり、ワードプロセッサ用ソフトウェアであるアプリケーションAP1および表計算用ソフトウェアであるアプリケーションAP2の識別情報である。つまり、当該ライセンスキーコードが装着されたPCでは、ライセンス管理装置Aの管理対象である3種類のアプリケーションのうち、アプリケーションAP1、AP2については使用が認められており、データベース管理用ソフトウェアであるアプリケーションAP3については使用が認められないことを示す。

【0024】なお、アプリケーションAP1、AP2、AP3には、図20に示した従来技術にあるようなトリガ命令は埋め込まれていない。ただし、アプリケーションAP1～AP3は起動時および停止時に、自身のアプリケーションID(ライセンスキーコードがライセンスデータ212として保持しているアプリケーションIDと同じもの)、プロセス間通信用のポート番号およびプロセスIDをライセンスマネージャ30に通知する。

【0025】ライセンスマネージャ30は、アプリケーションAP1～AP3についてライセンスチェックを行う。ライセンスマネージャ30によるライセンスチェックは、少なくとも5分に一度を目安として行われる。ライセンスマネージャ30は、OS11上の管理対象アプリケーションの稼動状況を把握しておき、上記のライセンス情報210の内容をもとに、使用が認可されていない管理対象アプリケーションが1つでも稼動していれば、稼働中の全ての管理対象アプリケーションに対して停止命令を出す。

【0026】次いで、ライセンスマネージャ30の構成と詳しい処理内容とについて説明する。

(ライセンスマネージャの構成) 図3は、ライセンスマネージャ30の構成を示すブロック図である。ライセンスマネージャ30は、スケジューラ31、検証部32、プロセス通信部33、停止命令発行部34、RAM35などを有する。

【0027】スケジューラ31は、ライセンスチェックの実行タイミングを決定し、検証部32に対し、ライセンスチェックの実行を指示する。スケジューラ31がライセンスチェックの実行を検証部32に指示するタイミングは、管理対象アプリケーションの起動時、および、内蔵タイマ311のタイムアップ時である。

【0028】管理対象アプリケーションの起動については、プロセス間通信部33経由で管理対象アプリケーション自身からスケジューラ31に通知される。内蔵タイマ311によるライセンスチェックの実行タイミング決定の手順は以下の通りである。スケジューラ31は、まず、ライセンスマネージャ30の起動時(=PC1の起動時)にタイマ311を起動してカウントを開始させ、その後、5分間が経過してタイマ311がタイムアップするたびに、検証部32にライセンスチェック実行を指示する。そして、検証部32からライセンスチェック実行終了の通知を受けると、タイマ311をクリアしたうえでカウントを0から再開させる。ただし、管理対象アプリケーションから起動通知を受け取った場合には、スケジューラ31は、タイマ311がタイムアップしたかどうかに関わりなく、検証部32にライセンスチェック実行の指示を出し、チェックが終了するとタイマをクリアしたうえでカウントを0から再開させる。これにより、最長でも5分の間隔でライセンスチェックが行われることになる。

【0029】検証部32は、スケジューラ31からの指示を受けてライセンスチェックを行い、ライセンスキーコードが保持しているライセンス情報210と、PC1における管理対象アプリケーションAP1、AP2、AP3の稼動状況とをもとに、ライセンスに違反する不正使用(装着されたライセンスキーコードが使用を認可していない管理対象アプリケーションが稼働中である、またはライセンスキーコードが装着されていない状態で管理対象アプリケーションが稼働している)を検出する。

ケーションが稼働中である)の有無を検証する。そして、検証部32は、不正使用を検知した場合、停止命令発行部33に対し、管理対象アプリケーションの停止処理実行を指示する。

【0030】検証部32は、ライセンスチェックにあたって2種類の情報を参照する。1つはライセンス情報210をもとに生成される使用可否情報であり、もう1つは稼動情報(PCにおける管理対象アプリケーションの稼動状況を示す情報)である。使用可否情報は、検証部32がライセンスキー200の保持するライセンス情報210を読み出し、そのままRAM35に格納したものである。検証部32は、使用可否情報の生成を、ライセンスマネージャが起動した時(=PC1起動時)、および、スケジューラ31からのライセンスチェック実行指示を受け付けた時に実行する。

【0031】図4は使用可否情報400の例を示す。図4(a)に示す使用可否情報400は、ライセンスキー200の保持するライセンス情報210が、そのままコピーされたものなので、その構成および内容はライセンス情報210と同じである。すなわち、キー識別番号410とライセンステータ420とから成る。各情報の内容の説明は、ここでは繰り返さない。

【0032】同図(a)では管理対象アプリケーションAP1、AP2、AP3のうち、ライセンス情報200においてライセンステータ212が存在していた管理対象アプリケーションAP1、AP2について、ライセンステータ420が存在している。なお、ライセンスキー200が装着されておらず、ライセンスキー200からのライセンス情報読み出しができない場合でも、検証部32は使用可否情報を生成するが、その場合の使用可否情報は、特別な内容となる。

【0033】同図(b)が、ライセンスキー200が装着されていない場合に生成される特別な使用可否情報430である。使用可否情報430においては、ライセンスキーID410の内容がNULL値とされ、同様に内容がNULL値であるライセンステータ420が1件だけ生成される。ライセンスキー200が装着されていない場合にも使用可否情報を生成するのは、ライセンスキー200が装着されていなくとも、管理対象アプリケーションが起動していなければ不正使用にはならないためである。そこで、ライセンスキー200が装着されていない場合には、1つでも管理対象アプリケーションが起動していれば「不正使用」と判定される形の使用可否情報430を生成しておくのである。

【0034】また、検証部32はライセンス情報210をライセンスキー200から読み出す前に、ライセンスキーID211をチェックし、RAM35に格納されている使用可否情報のライセンスキーID410と一致するかチェックする。一致した場合は、前回のライセンスチェック時にライセンス情報を読み出したライセンスキー

一がそのまま装着されている(ライセンスキーの付け替えは行われていない)ので、ライセンス情報を読み出して使用可否情報を生成することはせず、すでにRAM35に入っている使用可否情報をそのまま使用する。これにより不要な処理を省略できる。

【0035】稼動情報は、管理対象アプリケーションが起動時および終了時送信してくる情報(アプリケーションID、ポート番号、プロセスID)をもとに検証部32が生成し、RAM35に格納する。図5は、稼動情報の構成とその内容の一例(管理対象アプリケーションAP1、AP3が稼働中の場合)を示す。

【0036】稼動情報500は、アプリケーションID欄510、ポート番号欄520、プロセスID欄530からなる。アプリケーションID欄510には、どの管理対象アプリケーションに関する稼動情報であるかを示すアプリケーションIDが格納される。このアプリケーションIDは、ライセンス情報210および使用可否情報400に含まれるアプリケーションID(ライセンスデータ211、410)と同じものである。

【0037】ポート番号欄520には、管理対象アプリケーションが持つ固有のポート番号(プロセス間通信に用いられるポートのアドレス情報)が格納される。なお、ポートとはプロセス間通信において、通信を行うプロセスが送受信するデータを一時的に記憶させておくための共有データ領域である。プロセスID欄530には、OS11が起動時の管理対象アプリケーションに対してプロセス管理のために付加したプロセスID(プロセスの識別情報)が格納される。

【0038】稼動情報500に含まれる上記3種類の情報は、いずれも、管理対象アプリケーションが、起動の際にライセンスマネージャ30に通知してくるものであり、検証部32は、これら情報をプロセス通信部33経由で取得して稼動情報500を生成し、RAM35に格納する。また、検証部32は、管理対象アプリケーションが終了(ユーザの操作による正常終了または不正使用が発覚した際のライセンスマネージャ30による強制終了)した際には、これに対応する稼動情報をRAM35から消去する。

【0039】検証部32は、上記稼動情報と使用可否情報とからライセンスチェックを行う。具体的には、稼動情報を1件ずつ読み出し、当該稼動情報に含まれる管理対象アプリケーションのアプリケーションIDが、ライセンステータ420として使用可否情報400に含まれているかチェックする。使用可否情報にアプリケーションIDが含まれていなければ、これは、ライセンスキー200によって使用が認可されていない管理対象アプリケーションがPC1上で稼動していることを意味するので、検証部32は不正使用と判定する。なお、ライセンスキー200が装着されていない場合の使用可否情報430には、ライセンステータ420として格納されてい

るアプリケーションIDが存在しないので、管理対象アプリケーションが1つでも稼動していれば（稼動情報が1件でも存在すれば）、検証部32は必ず不正使用と判定することになる。

【0040】停止命令発行部33は、不正使用を検知した検証部32からの指示に従い、稼働中の管理対象アプリケーション全てに対して停止命令を発行し、管理対象アプリケーションからプロセス終了の通知を待ち、停止処理の完了を検証部32、スケジューラ31に通知する。なお、停止命令発行部34は、停止命令を発行する際には、RAM35内の稼動情報500を参照し、通信ポート欄520に格納されたポート番号に宛てて停止命令を送信する。

【0041】プロセス通信部33は、ライセンスマネージャ30と他のプロセス(OS11、管理対象アプリケーションAP1、AP2、AP3)とのプロセス間通信を行う。なお、プロセス通信部33が管理対象アプリケーションとの間で行う通信はコネクション型である。プロセス通信部33は、RAM35内の稼動情報500に含まれるポート番号を用いて管理対象アプリケーションとの通信を確立して、情報のやりとりを行う。プロセス通信部33に対しては、管理対象アプリケーションがユーザ操作によって起動／終了した際に、その旨を通知してくれる。プロセス通信部33はこれを受けて、管理対象アプリケーションの起動／終了を検証部32に通知する。

【0042】RAM35には、ライセンスマネージャ30がライセンスチェック処理に用いる情報、すなわち、使用可否情報400および稼動情報500が格納されている。

(ライセンスマネージャの動作)以下、本実施の形態におけるライセンスマネージャ30各部が、ライセンスチェックにともなって実行する動作を、フローチャートに従って説明する。

【0043】図6は、ライセンスマネージャ30の動作を示すフローチャートである。まず、PC1の電源がONとなった時点で、ライセンスマネージャ30はOS11の子プロセスとして起動される。すると、スケジューラ31が検証部32に対しライセンキー200のチェックを指示する(S601)。検証部32はライセンスキー200にアクセスして、ライセンスキー200の保持するライセンス情報を読み出し、これをもとに使用可否情報(図4(a))を生成してRAM35に格納する

(S602)。キーが装着されていなかった場合は、全ての管理対象アプリケーションについて「使用不可」となる形の特別な使用可否情報430(図4(b))を生成する。

【0044】スケジューラ31は、検証部32から使用可否情報の生成完了の通知を受けると、内蔵のタイマ311を起動する(S603)。その後、スケジューラ31は、

1は、プロセス通信部33経由で管理対象アプリケーションから起動の通知を受けるか(S605:Yes)、タイマ311がタイムアップ(5分経過)するか(S604:Yes)した時点で、検証部32にライセンスチェック処理実行を指示する。

【0045】検証部32はスケジューラ31からの指示を受けて処理を開始し、管理対象アプリケーション起動の場合(S605:Yes)は、当該アプリケーション用の稼動情報を生成した上で(S606)、タイムアップの場合(S604:Yes)はそのまま、使用可否情報と稼動情報を照合して不正使用の有無を検証する。その際、検証部32は先ず、ライセンスキー200にアクセスする(S607)。そして、ライセンスキーIDの情報をもとに、前回のライセンスチェック時と同じライセンスキーが装着されているかどうか判定する。キーに変更があった場合(S608:Yes)、キーが取り替えられたのであれば、取り替えられたキー(その時点で装着されているキー)が保持しているライセンス情報をもとに新しい使用可否情報を生成し、キーが取り外されていた場合は、キーIDがNULL値で、全ての管理対象アプリケーションについて「使用不可」とした使用可否情報を生成する(S609)。前回と同じキーが装着されれば(S608>No)、その時点でRAM35に格納されている使用可否情報をそのまま使用する。

【0046】次いで、検証部32は、使用可否情報と稼動情報を照合し、ライセンス情報をにおいて使用不可とされているアプリケーションが稼動していないかチェックし、稼動していれば(S610:Yes)、停止命令発行部34に稼働中の管理対象アプリケーションのポート番号を通知する。停止命令発行部34は停止命令を生成すると、これをポート番号とともにプロセス通信部33に出力して、当該ポートに宛てて停止命令を送信させる(S611)。その後、停止命令発行部34は、プロセス通信部33経由で管理対象アプリケーションから終了通知が届くのを待ち(S612:Yes)、これを検証部32に通知する。検証部32は、RAM35内の稼動情報をクリアする(S613)。そして、検証部32はスケジューラ31に指示してタイマ311の内容をクリアさせたうえで(S614)、ステップS603(タイマ311の再起動)に戻る。

【0047】また、不正使用が検知されなかった場合には(S610>No)、検証部32はスケジューラ31に指示してタイマ311の内容をクリアさせたうえで(S614)、ステップS603に戻る。そして、ライセンスマネージャ30は、ステップS603～S614の処理をPC1の電源がOFFになるまで繰り返す(S615)。

【0048】なお、図6においては、ステップS605の管理対象アプリケーションの起動は、割り込み的に発生するので、フローチャートに示すタイミング(ステッ

PS 603とS 607との間)で発生するとは限らない。このタイミング以外のタイミングでアプリケーション起動が発生した場合は、処理がステップS 603に戻るまで、スケジューラ31が保留しておき、その後、ステップS 605の処理を実行する。

【0049】上記の通り、本実施の形態におけるライセンス管理装置は、管理対象アプリケーションとは別個のプロセスとして動作するライセンスマネージャが、ライセンスキーの有無およびライセンスキーが保持するとライセンス情報と、管理対象アプリケーションの稼動状況とをもとに、少なくとも所定時間(5分)に1度は不正使用の有無を確認するライセンスチェックを行う。これにより、個々の管理対象アプリケーションにトリガ命令(ライセンスチェック用命令)を埋め込む必要がなくなり、アプリケーションの肥大化を防止できる。また、管理対象アプリケーションの数に関わりなく、ライセンスチェックをライセンスマネージャがまとめて行うので、管理対象アプリケーションの数が増えても、ライセンスチェック処理の負荷が大きくなることはないので、アプリケーションやシステムのパフォーマンス低下を招くことなく、管理対象アプリケーションの不正使用を防止できる。また、上記実施の形態においては、ライセンスマネージャと管理対象アプリケーションとは、共にOSのネーブルとして、ライセンスマネージャは管理対象アプリケーションの起動には関係せず、停止させる際にも停止命令を送信するのみであるため、ライセンスマネージャの負荷は小さく、システムのパフォーマンスへの影響も小さい。

【0050】なお、ライセンスキーは1種類の管理対象アプリケーションしか管理しない場合もある。その場合、ライセンスマネージャは当該管理対象アプリケーションが起動しているかどうか、および、ライセンスキーが装着されているかどうか、をもとに不正使用を判定する。その場合、ライセンス情報(図2)のうちライセンスデータは不要となり、稼動データについても、不正使用検知時の停止命令送信先であるポート番号のデータだけを保持しておけばよいので、その分ライセンスマネージャによる不正使用の判定処理は簡単になる。

【0051】また、上記実施の形態と同様、ライセンスキーが複数種類の管理対象アプリケーションを対象としても、一部アプリケーションは使用可能で一部は使用不可という形のライセンス形式でなく、当該ライセンスキーが装着されれば全ての管理対象アプリケーションが使用可能という形であれば、1種類のアプリケーションのみをライセンス管理の対象とする場合と同様、ライセンス情報(図2)のうちライセンスデータは不要となり、稼動データについても、不正使用検知時の停止命令送信先であるポート番号のデータだけを保持しておけばよいので、その分ライセンスマネージャによる不正使用の判定処理は簡単になる。

【0052】また、本実施の形態では、不正使用検知時には管理対象アプリケーションに停止命令を送信するましたが、プログラムの改ざなどによりアプリケーションが停止命令に応じなかった場合には、OSにプロセスIDを通知して、OSによる強制終了を要求することにしてもよい。そして、本実施の形態では、不正使用を検知した場合、稼働中の管理対象アプリケーションを全て停止させることとしたが、不正使用されている管理対象アプリケーションに限って停止させることにしてもよい。

【0053】(変形例1) 本変形例は、ライセンスマネージャが、ライセンスチェック実行の前にライセンスマネージャと管理対象アプリケーション以外のプロセス(管理対象外のアプリケーション、データ入出力など)の状況をOSに問い合わせ、他のプロセスがアイドル状態にあるタイミングを選んでライセンスチェックを行うものである。

【0054】本変形例におけるライセンスマネージャの構成は、前記実施の形態におけるライセンスマネージャ30の構成と同じであり、検証部およびプロセス通信部の動作が部分的に異なるのみである。よって構成図は特に示さない。ただし、検証部およびプロセス通信部については、名称が同じでも動作が異なるので、以下の説明では実施の形態におけるライセンスマネージャ30、検証部32、プロセス通信部33と区別するために、参照番号を変えてライセンスマネージャ301、検証部321、プロセス通信部331とする。

【0055】図7は、本変形例におけるライセンスマネージャ301の動作を示すフローチャートである。図6に示した前記実施の形態におけるライセンスマネージャ30の動作とほとんど同一であり、共通の処理については、同じステップ番号を付加し、説明は省略する。検証部321は、スケジューラからの指示に応じて不正使用の有無を検証する処理(ステップS 604: YesまたはS 605: Yes)を開始しようとする場合、その前に、プロセス通信部331経由でOS 11に他プロセスの実行状況を問合せ、ライセンスマネージャ301および管理対象アプリケーションAP1～AP3以外のプロセスがアイドル状態となるのを待って(S 701: Yes)、ライセンスチェックの処理(ステップS 607～S 614)を実行する。

【0056】OS上では、普通、ライセンスマネージャと管理対象アプリケーション以外のプロセス(管理対象外のアプリケーション、データ入出力など)が稼動しているが、上記の通り、本変形例によれば、それらのプロセスのパフォーマンスが、ライセンスチェック処理の影響で低下することを防止できる。ただし、他プロセスがアイドル状態となるまでの待ち時間があまりに長いと、不正使用の防止というライセンス管理本来の目的達成に支障をきたすおそれがある。そこで、タイマを用いて待

ち時間を監視し、タイムアップした場合は他プロセスがアイドル状態となっていなくてもライセンスチェック処理を実行することにしてもよい。

【0057】あるいは、OSから稼働中の他プロセスについて種類の情報まで取得し、プロセスの種類ごとに優先順位をつけ、優先順位の高いプロセスに限って、アイドル状態となるのを待つことにもよい。

(変形例2) 本変形例は、不正使用を検知したライセンスマネージャが、管理対象アプリケーションに停止命令を出す際、管理対象アプリケーションに対してコールバック機能の使用を行うかどうかを確認し、コールバック機能の使用を求める応答をした管理対象アプリケーションに対しては、コールバック機能の使用を認めるものである。コールバック機能を使用する管理対象アプリケーションは、起動時に、コールバック機能を使用することをライセンスマネージャに通知し、ライセンスマネージャは、コールバック機能を使用する管理対象アプリケーションのアプリケーションIDを稼働情報と併せて保持しておく。

【0058】ここでいうコールバック機能とは、アプリケーションが停止時の後処理を行うための機能であり、例えば、データベース管理ソフトウェアなどを処理途中で強制的に終了させる場合、終了時にデータベース整合性保持のための後処理を行わないと、データベースの内容に不整合が生じてしまう。そこで、コールバック機能を利用して、整合性保持のための後処理を行うのである。

【0059】図8は、不正使用検知にともなう管理対象アプリケーション停止時に、ライセンスマネージャ302、管理対象アプリケーション、ライセンスキーワードの間で交わされるメッセージシーケンスを示す図である。同図(a)は、管理対象アプリケーションがコールバック機能を使用する場合であり、同図(b)は、管理対象アプリケーションがコールバック機能を使用しない場合である。

【0060】同図(b)の場合、不正使用を検知したライセンスマネージャ302は、管理対象アプリケーションに停止命令を送るのみであるが、同図(a)の場合、ライセンスマネージャ302は、稼働情報とともに保持していた情報(コールバック機能を使用する管理対象アプリケーションのアプリケーションID)から、この管理対象アプリケーションがコールバック機能を使用することが分かるので、停止命令の送信とともにコールバック機能使用を認可する。管理対象アプリケーションは、コールバック機能を用いて終了時の後処理を行った後に停止する。

【0061】このように、本変形例によれば、不正使用検知時に管理対象アプリケーションを停止させる場合でも、必要な後処理が実行されるので、その後の管理対象アプリケーションの使用に不都合が生じることはない。

(変形例3) 本変形例は、ライセンスマネージャが、管理対象アプリケーションをみずから終了させる権限を有するものである。これにより、管理対象アプリケーションの不正使用に対する処理を厳しく行うことができる。

【0062】前記実施の形態で、不正検知時にライセンスマネージャ30が管理対象アプリケーションに対して行うのは「停止命令の送信(終了を指示すること)」であり、アプリケーションの停止処理は管理対象アプリケーションが自身の処理として行うため、極端な例では、管理対象アプリケーションの中の「マネージャからの終了指示に応じて停止する」部分あるいは「起動したこと」をライセンスマネージャに通知する部分のプログラムを削除するなどの改ざんが加えられると、不正使用の管理対象アプリケーションの停止ができなかつたり、あるいはそれ以前の問題として起動したことすら検知できないこともあります。

【0063】図9は、本変形例におけるライセンスマネージャ303と管理対象アプリケーションAP1～AP3との関係を示す階層図である。前記実施の形態では、ライセンスマネージャ30と管理対象アプリケーションAP1～AP3とは図1で示したように、いずれもOS11の子プロセスとして動作していたのに対し、本変形例では、管理対象アプリケーションAP1～AP3をライセンスマネージャ303の子プロセスとしている。親子関係となることで、ライセンスマネージャ303は、前記実施の形態ではOS11が行っていた管理対象アプリケーションAP1～AP3の起動／終了を、みずから管理することができる。

【0064】図10は、本変形例のライセンスマネージャ303の構成を示すブロック図である。ライセンスマネージャ30とは、停止命令発行部33の代わりに子プロセス管理部1001を有する点および、プロセス通信部333と検証部323が固有の処理を行う点で異なる。子プロセス管理部1001は、ユーザからの管理対象アプリケーションの起動指示をプロセス通信部333経由で受け付けると、この管理対象アプリケーションを子プロセスとして起動させる。子プロセス管理部1001は、起動させた管理対象アプリケーションにみずからが付加したプロセスIDと管理対象アプリケーションのアプリケーションIDとを検証部323に通知して稼働情報を生成させるとともに、管理対象アプリケーションの起動をスケジューラ31にも通知する。そして、スケジューラ32からの指示に応じたタイミングで検証部323がライセンスチェックを行って不正使用と判定した場合、子プロセス管理部1001は、この判定結果の通知を受けて、子プロセスとして稼働中の管理対象アプリケーションを全て停止させる。

【0065】検証部323が行う処理は、前記実施の形態の検証部32が行う処理とほぼ同じである。ただ、検証部323が生成する稼働情報には、ポート番号(図5

参照)の情報は含まれない。子プロセス管理部1001が、管理対象アプリケーションを直接管理し、これらを停止させる際も、停止命令の送信という間接的なやり方でなく、親プロセスとしての権限で直接に停止させるため、停止命令の送信先としてポート番号を保持しておく必要がないからである。

【0066】プロセス通信部333は、OS11との通信を行い、管理対象アプリケーションとの通信(管理対象アプリケーションからの起動通知、管理対象アプリケーションへの停止命令送信)は行わない(子プロセス管理部1001が行うため)。このように、ライセンスマネージャが管理対象アプリケーションを子プロセスとして管理する方法をとれば、管理対象アプリケーションの起動を確実に把握できるとともに、不正使用検知時にも間違いなく停止させることができるので、強力なライセンス管理が可能となる。

【0067】なお、ここでは、子プロセス管理部1001は、管理対象アプリケーションをいったん起動させた後に、検証部323にライセンスチェックを行わせているが、管理対象アプリケーション起動前に、当該管理対象アプリケーションのアプリケーションIDを検証部323に通知して、ライセンス情報をもとに「起動の可否」を判定させてもよい。こうして起動前に不正使用となるかどうか判定すれば、そもそも使用できないアプリケーションを起動させるという無駄な処理が不要となる。

【0068】また、上記の方法では、管理対象アプリケーションの起動・停止をすべて親プロセスたるライセンスマネージャ303が処理するので、ライセンスマネージャ303の負荷は大きく、これはシステムのパフォーマンス低下の防止という観点からは望ましくない。さらに、管理対象アプリケーションのバージョンアップや管理対象アプリケーションの追加などあった場合、ライセンスマネージャ303が管理対象アプリケーションを子プロセスとして起動するために保持している情報に、こうした変更を反映するための更新処理(ライセンスマネージャの保守処理)が発生する。(こうした問題は、ライセンスマネージャと管理対象アプリケーションとを共にOSの子プロセスとしていた上記実施の形態では発生しない。)

そこで、上記実施の形態よりも停止処理を厳密に行え、しかもライセンスマネージャの負荷が比較的軽くてすむ方法として、ライセンスマネージャと管理対象アプリケーションとを共にOSの子プロセスとして動作させつつ、ライセンスマネージャにシステム権限を付加する方法がある。この方法であれば、ライセンスマネージャは、負荷の増大や保守作業の発生なしに、管理対象アプリケーションを自らの手で確実に停止させることができる。

【0069】ここで言うシステム権限とは、OS上で独

立して(親子関係なしに)動作している他のプロセスを終了させる権限という意味である。システム権限を付加されたマネージャは、不正使用検知時、この権限に基づいて管理対象アプリケーションを終了させることができる。停止にあたっては、稼動情報に含まれるプロセスIDをもとに停止処理を行う。システム権限は、ライセンスマネージャがOSに要求し、OSがこれに応じてライセンスマネージャに付加する。

【0070】このように、システム権限を付加する方法をとれば、ライセンスマネージャは管理対象アプリケーションの起動処理を行うことなく、停止だけを行うことができる。その分ライセンスマネージャの負荷は軽くてすみ、管理対象アプリケーションのバージョンアップや追加に伴なう保守処理も不要である。

(変形例4) 本変形例では、ライセンスマネージャが同報通信手段を有し、不正使用を検知した際の管理対象アプリケーションへの停止命令の通信を同報通信で行うこととしてもよい。ここでいう同報通信とは、コネクションレス通信で複数の相手に同一のデータを送信することである。同報通信の場合、ライセンスマネージャが同報通信手段を有すると共に、管理対象アプリケーションの側も同報通信用の通信ポートから通知を受け取る手段を有することが前提となる。また、同報通信の場合、管理対象アプリケーションは固有のポート番号を有することなく、あらかじめ定められた単一の同報通信用ポートで通知を受信する。

【0071】前記実施の形態では、ライセンスマネージャと管理対象アプリケーションとはコネクション型接続で通信を行っていた。そのため、不正使用を検知して全ての管理対象アプリケーションに停止命令を送信する場合も、ライセンスマネージャは、1つ1つの管理対象アプリケーションと順次コネクション型接続を確立した後、停止命令を送信する動作を繰り返していた。

【0072】しかるに、本変形例におけるライセンスマネージャは、同報通信手段を有することで、不正使用を検知して管理対象アプリケーションに停止命令を発行する際、個々に固有のポート番号を有する管理対象アプリケーション(通信ポート)との間の通信を別々に確立することなく、あらかじめ定められた単一の同報通信用(管理対象アプリケーション共通)のポート番号に対して停止命令を送信する。すると、一度の同報通信で全ての管理対象アプリケーションへ停止命令を送信できる。そのため、停止命令の送信処理におけるライセンスマネージャの負荷が小さくてすみ、ライセンス管理処理の負荷によりシステムのパフォーマンスが低下するおそれは小さくなる。

(変形例5) 本変形例は、本発明のライセンス管理装置をネットワークで接続された複数のPCに適用するものである。

【0073】図11は、本変形例のライセンス管理装置

A5を、ネットワーク1100上の複数のPC上で動作させるネットワークシステムを示す。ネットワーク1100には、1台のサーバPC1110と複数のクライアントPC1120a～1120nが接続されている。サーバPC1110および複数のクライアントPC1120a～1120nには、識別情報(PCごとのアドレス)が付加されており、サーバPC1110の識別情報は" S-PC "、クライアントPC1120a～1120nの識別情報は、それぞれ" C-PCa "～" C-Pc "である。

【0074】ライセンスキ-205は、サーバPC1110に装着されており、ライセンスマネージャ305は、サーバPC1110上で稼動する。管理対象アプリケーションAP1～AP3は、クライアントPC1120a～1120nで稼動する。本変形例における、ネットワーク1100上でのライセンスチェックは、ライセンスマネージャ305と管理対象アプリケーションとのやりとりがネットワーク1100を介して行われる点、ライセンスチェックに必要な管理対象アプリケーションの稼動情報がクライアントPC1120a～1120nからネットワーク1100を経由してサーバPC1110上のライセンスマネージャ305に送信される点、さらに、ライセンスマネージャ305がチェックする点が、ネットワーク1100上で管理対象アプリケーションを使用できるユーザ数である点が異なっている。また、ライセンスキ-205が保持するライセンス情報の内容も、ネットワーク用の内容となる。

【0075】図12は、本変形例においてライセンスキ-205が保持するライセンス情報1200の内容である。ライセンス情報1200は、1つのライセンスキ-識別情報1210、認可ユーザ情報1220とからなる。認可ユーザ情報1220は、管理対象アプリケーションの数と同数であり、アプリケーションID1221および使用可能ユーザ数1222とから成る。アプリケーションID1221は、当該ライセンス情報がどの管理対象アプリケーションに対応するかを示すもので、使用可能ユーザ数1222は、当該ライセンスキ-が使用されるネットワーク上で、アプリケーションID1221に示す管理対象アプリケーションを同時に使用できるユーザ数を示す。

【0076】図12に示すライセンス情報1200は、ライセンスキ-205が使用されるネットワーク上では、アプリケーションAP1を5人のユーザが、アプリケーションAP2を4人のユーザが、アプリケーションAP3を1人のユーザが、同時に使用できることを示している。ライセンスマネージャ305はサーバPC1110上で動作しながら、クライアントPC1120a～1120n上で動作する管理対象アプリケーションAP1～AP3のライセンスをチェックする。

【0077】先ず、ライセンスマネージャ305は、ラ

イセンスキ-205からライセンス情報1200を読み出し、これをそのまま使用可否情報として格納する。次いで、クライアントPC1120a～PC1120nから送信されてくる管理対象アプリケーションの起動／終了通知をもとに稼動情報を生成する。図13は、本変形例における稼動情報1300を示す。稼動情報1300は、アプリケーションID欄1310、稼動ユーザ数欄1320、ポートID欄1330、プロセスID欄1340から成り、ポートID欄1330およびプロセスID欄1340には、ユーザ数に相当する数だけのポートIDおよびプロセスIDのデータが格納される。さらに、ポートIDおよびプロセスIDには、どのクライアントPCで稼働中のアプリケーションに対応するかを示すため、ネットワーク1100におけるクライアントPCの識別情報(図11参照)が付加される。

【0078】ライセンスマネージャ305は、ライセンス情報1200における使用可能ユーザ数の値と稼動情報に含まれる稼動ユーザの値とから不正使用の有無を判定し、不正使用と判定した場合は、稼動情報1300に含まれるポート番号のデータをもとに、ネットワーク1100上で稼働中の管理対象アプリケーション全てに停止命令を出す。

【0079】本変形例におけるライセンスマネージャ305の構成は、前記実施の形態におけるライセンスマネージャ30の構成と同じであり、検証部の動作が異なるのみである。よって構成図は特に示さず、以下、フローチャートを用いて動作のみ説明する。ただし、検証部については、名称が同じでも動作が異なるので、以下の動作の説明では、実施の形態における検証部32と区別するために、参照番号を変えて検証部325とする。

【0080】図14は、本変形例におけるライセンスマネージャ305の動作を示すフローチャートである。図6に示した前記実施の形態におけるライセンスマネージャ30の動作とほとんど同一であり、共通の処理については、同じステップ番号を付加し、説明は省略する。先ず、ライセンスマネージャ305の起動後、検証部325は、ライセンスキ-205からライセンス情報1200を読み出し(S601)、これをそのまま使用可否情報として格納する(S1401)。そして、検証部325は、ネットワーク1100およびスケジューラ31経由でクライアントPC上で管理対象アプリケーションから起動の通知を受けると(S605:Yes)、稼動情報1300のうち当該アプリケーション用の稼動ユーザ数欄1320の値をカウントアップし、管理対象アプリケーションのポート番号とプロセスIDとを稼動情報に格納した上で(S1402)、ライセンスチェック処理(ステップS607以下)を開始する。なお、フローチャートに図示はしていないが、クライアントPCは管理対象アプリケーションの終了もライセンスマネージャ305に通知する。その場合、検証部325は、対応する

稼動ユーザ数欄1320の値を1つ減らし、管理対象アプリケーションのポート番号とプロセスIDとを稼動情報から削除することで、管理対象アプリケーションの稼働状態を正確に把握する。

【0081】ライセンスチェック処理（ステップS607以下）において、検証部325は、管理対象アプリケーションについて順次、対応する使用可能ユーザ数1220（ライセンス情報1200内）と稼動ユーザ数（稼動情報1300内）とを比較し、1つでも稼動ユーザ数が使用可能ユーザ数を上回っていれば、不正使用と判定する（S1404）。

【0082】このように、本変形例によれば、ネットワーク上で稼動するシステムにおいても、アプリケーションの肥大や、システムおよびアプリケーションのパフォーマンス低下を招くことなくライセンス管理を実行できる。なお、本変形例でも、上記実施の形態と同様、1つでも不正使用と判定された管理対象アプリケーションがあれば、全ての管理対象アプリケーションを終了させることとしているが、ネットワーク上の稼動ユーザ数を使用者が常に正確に把握しているのはかなり困難である。不注意によって稼動ユーザ数が許された値を越える可能性もある。よって、不正使用と判定された管理対象アプリケーションに限って停止させることにしてよい。その場合、全てのユーザ（クライアントPC）について当該管理対象アプリケーションを停止させるやり方と、最後に当該管理対象アプリケーションを起動したユーザ（使用可能ユーザ数が4の場合、5番目に起動したユーザ）に限って停止させるやり方が考えられる。後者のやり方の場合、検証部は、稼動情報の中に起動時刻を格納しておき、これをもとに最後に起動したユーザを判定すればよい。

（変形例6）本変形例は、ライセンス管理装置がライセンスキーの着脱を監視するものである。前記実施の形態においては、ライセンスキーの取り外しや付け替えによる不正使用は、一定の時間間隔において、または、管理対象アプリケーションの起動時に、ライセンスマネージャからライセンスキーにアクセスに行って確認していたのに対し、本変形例では、ライセンスキーが外されると即座にこれを検知できるので厳密なライセンス管理が可能となる。

【0083】図15は、本変形例におけるライセンス管理装置Bの構成を示す図である。ライセンス管理装置Bは、ライセンスキー200およびライセンスマネージャ306に加えて、PC1Bのコネクタへのライセンスキー200の装着を監視するキーチェッカー1500を更に備える。ライセンスマネージャ306の構成は、前記実施の形態におけるライセンスマネージャ30（図3）と同じなので、構成図は示さない。ただし、ライセンスマネージャ306における検証部および停止命令発行部の処理はライセンスマネージャ30におけるものと一部

異なるので、区別のために、以下、参照番号を変えて、検証部326、停止命令発行部346と記す。

【0084】PC1Bのソフトウェア層10Bでは、セキュリティマネージャ306および管理対象アプリケーションAP1～AP3とともに、キーチェッカー1500が稼動している。キーチェッカー1500はハードウェア層20Bにおけるライセンスキー200とPC1Bとの接点であるコネクタを監視している。キーチェッカー1500は、ソフトウェアであり、ライセンスマネージャ30などと共に媒体（CD-ROM）に格納された形で配布され、ライセンスマネージャ30とともにPC1Bにインストールされたものである。キーチェッカー1500は、ライセンスマネージャ306起動後に、検証部326によって、ライセンスマネージャ306の子プロセスとして起動される。

【0085】コネクタを監視していたキーチェッcker1500は、ライセンスキー200が取り外されると、これを検知してライセンスマネージャ306（検証部326）に割込み信号を送信する。検証部326は、使用可否情報を、全ての管理対象アプリケーションについて「使用不可」とする内容（図4参照）に変更した上でライセンスチェックを実行する。ライセンスチェックの結果、稼働中の管理対象アプリケーションが1つでもあれば不正使用と判定され、停止命令発行部346は、稼働中の管理対象アプリケーション全てに停止命令を出す。また、これと同時に、停止命令発行部346は、不正使用検知によるアプリケーション強制停止を通知するメッセージをPC1Bのディスプレイ上に表示させる。ただし、ライセンスキーが取り外されても、管理対象アプリケーションが1つも稼動していない状態では不正使用にはならないことは言うまでもない。

【0086】図16は、停止命令発行部346が出力せる、強制停止の通知メッセージの一例である。さらに、キーチェッcker1500は、いったん取り外されたライセンスキー200が再装着された場合、再装着を通じる割込み信号をライセンスマネージャ306（検証部326）に送信する。検証部326は、これに対し、ライセンスキー200のライセンス情報を読み出して使用可否情報を作り直す。これによって、いったん強制終了させられた管理対象アプリケーションの再起動が可能となる。

【0087】このように、本変形例によれば、ライセンスキーの着脱をその時点で検知できるので、極めて厳密なライセンスチェックが可能となる。なお、上記の方式では、ライセンスキーが偶然あるいは不注意により装着されたコネクタから外れてしまった場合、あるいは、ユーザ間でライセンスに違反しない形でライセンスキーの交換を行う場合にも不正使用と判定されて、稼働中の管理対象のアプリケーションが全て停止されてしまうことになる。すると、ユーザとしては使い勝手が悪いと感じ

るかもしれない。そこで、ライセンスマネージャがキー チェッカーからの割り込み信号によりライセンスキーの取り外しを検知した場合、ライセンスマネージャは警告のメッセージをPCのディスプレイ上に表示させ、ある程度の猶予時間が経過してもライセンスキーが正しく再装着されなかつた場合に限りライセンスチェックを行い、不正使用の管理対象アプリケーションを停止させることにしてもよい。

【0088】また、本変形例においては、ライセンスマネージャは、前記実施の形態においてタイム311を用いて一定の時間間隔で行っていたライセンスチェックを行わないことにもよい。このタイム311を用いたライセンスチェックは、稼働中のライセンスキーの取り外しおよび付け替えによる不正使用を想定したものである。本変形例では、取り外しや付け替えはキーチェッカーで検知できるので、キーチェッカーからの通知があつた場合にライセンスチェックを行えば、上記の不正使用は検知できるからである。ライセンスマネージャが一定の時間間隔でライセンスキーにアクセスする必要がなくすれば、ライセンスマネージャの負荷は低減でき、システムのパフォーマンスに与える影響を小さくできる。

(変形例7) 本変形例は、販売促進を目的としたアプリケーションの試用にライセンス管理を適用するものである。

【0089】前記実施の形態のライセンスマネージャ30は、管理対象アプリケーションについて、ライセンスキーによって使用を認可されていない限り、稼働を無条件に禁止していたが、本変形例では、販売促進を目的として管理対象アプリケーションの試用を認める場合には、ライセンスキーなしでも、使用可能な機能を一部に限定した「体験版モード」で管理対象アプリケーションの稼働を認める。

【0090】ここでは、ライセンスキーにより使用が認められていない管理対象アプリケーションについても、作成したデータの保存（ファイル書き込み）を許さないという条件で試用を認めるという場合を例に説明する。本変形例におけるライセンスマネージャ307の構成は、前記実施の形態におけるライセンスマネージャ30（図3参照）と同じだが、検証部の動作が異なる。前記実施の形態におけるライセンスマネージャ30の検証部32と区別するために、参考番号を変えて検証部327と記す。以下に、ライセンスマネージャ307（検証部327）の動作を説明する。

【0091】図17は、本変形例におけるライセンスマネージャ307の動作を示すフローチャートである。前記実施の形態におけるライセンスマネージャ30の動作と共に通する部分には同じ参考番号を付し、説明は省略する。ライセンスマネージャ307の起動から、検証部327が使用可否情報と稼働情報をもとに不正使用のチェックを行うまで（ステップS601～S609）は、

ライセンスマネージャ30と同じである。

【0092】検証部327は、ライセンスキーによって使用が許可されていないにもかかわらず稼働中の管理対象アプリケーションがあった場合（S610：Yes）、更に、当該管理対象アプリケーションが試用可能なものかどうかチェックする（S1701）。検証部327は、試用可能な管理対象アプリケーションのアプリケーションIDを予め保持しているものとする。

【0093】当該管理対象アプリケーションが試用不可であった場合、検証部327は管理対象アプリケーション停止のための処理を行う（ステップS611～S613）。一方、試用可能な管理対象アプリケーションであった場合、検証部327はプロセス間通信部33経由で、当該管理対象アプリケーションに対して「体験版モード」での動作するよう指示する（S1702）。「体験版モード」では、管理対象アプリケーションは、例えば、「ファイル」－「保存」メニューを無効にするなどして、ファイル保存の指示を受け付けない形で動作する。「体験版モード」は、当該管理対象アプリケーションのプロセスが終了するまで有効となる。

【0094】よって、試用の結果、当該管理対象アプリケーションを正式に（ライセンス契約に基づいて）使用したいとユーザが考えた場合は、いったん当該管理対象アプリケーションを終了させ、ライセンス契約を結んで当該管理対象アプリケーションの使用を認めるライセンスキーを取得し、取得したライセンスキーをPCに装着した上で当該管理対象アプリケーションを再起動する、という手順をふむことで、当該管理対象アプリケーションを制限なしで使用できるようになる。

【0095】このように、本変形例によれば、不正使用を防止しつつ、販売促進を目的とした管理対象アプリケーションの試用も可能とすることができる。なお、ここではファイル書き込みを禁じる形での試用を許可する場合を例に説明したが、これは一例であり、生成したデータの保存またはプリント出力を禁止する（生成物を残せない）など、試用の形態は他にも考えられる。

【0096】

【発明の効果】以上の説明から明らかなように、本発明のライセンス管理装置は、ライセンス管理の対象となる1以上の管理対象アプリケーションプログラムについてコンピュータシステム上の使用状況を管理するライセンス管理装置であって、前記1以上の管理対象アプリケーションプログラムについて、前記コンピュータシステムでの使用の可否を示すライセンス情報を保持する、ハードウェアのライセンスキーと、前記コンピュータシステム上で、前記1以上の管理対象アプリケーションプログラムとは別のプロセスとして動作し、前記ライセンス情報をもとにライセンスチェックを行って前記1以上の管理対象アプリケーションプログラムの不正使用を防止するマネージャ手段とを備えることを特徴とする。よつて

て、管理対象アプリケーションが別個に内蔵されたトリガ命令を実行する形でライセンスチェックを行うことがないので、管理対象アプリケーションの肥大、さらにはアプリケーションソフトウェアやシステムのパフォーマンス低下を招くことなく、管理対象アプリケーションソフトウェアの不正使用を防止できる。

【0097】また、本発明のライセンス管理装置において、前記マネージャ手段は、前記ライセンスキーカーから前記ライセンス情報を読み出す読み出し手段と、前記1以上の管理対象アプリケーションプログラムの前記コンピュータシステム上の実行状況を示す稼動情報を取得する取得手段と、前記読み出し手段が読み出したライセンス情報と前記取得手段が取得した稼動情報をもとに、前記1以上の管理対象アプリケーションプログラムの中に、前記コンピュータシステムでの動作が認可されていないにもかかわらず動作しているものがあれば、不正使用ありと判定する判定手段と、前記判定手段が不正使用ありと判定した場合、稼働中である前記1以上の管理対象アプリケーションプログラムのうち、少なくとも前記判定手段によって前記コンピュータシステムでの動作が認可されていないにもかかわらず動作していると判定されたものを停止させるための処理を行う停止手段とを備える、とすることもでき、それによって、管理対象アプリケーションソフトウェアの不正使用を検知し、不正使用されている管理対象アプリケーションを停止させることが可能となる。

【0098】また、不正使用の検知について、前記読み取り手段は、ライセンスキーカーがシステムに存在しなければ、その旨を前記判定手段に通知し、前記判定手段は、ライセンスキーカーがシステムに存在しない場合、前記管理対象アプリケーションが1つ以上でも前記コンピュータシステムで稼動していれば不正使用ありと判定する、というやり方で簡単に行うこともできる。

【0099】また、本発明のライセンス管理装置は、前記マネージャ手段によるライセンスチェックの実行タイミングを制御する制御手段を更に備えることとしてもよく、それによって、ライセンスチェック処理は適当な時間間隔で実行され、不正使用を防止できる。また、本発明のライセンス管理装置は、前記コンピュータシステムにおける、前記1以上の管理対象アプリケーションプログラム以外のプロセスの稼動状況に関する情報を取得するプロセス情報取得手段を更に備え、前記制御手段は、前記稼動情報取得手段が取得した情報に基づいて、前記マネージャ手段によるライセンスチェックの実行タイミングを変更することとしてもよく、それによって、ユーザにとって重要な他のプロセスのパフォーマンスがライセンスチェックのために低下することを防止できる。

【0100】また、本発明のライセンス管理装置が管理する前記1以上の管理対象アプリケーションプログラムはコールバック機能を有し、前記停止手段は、前記1以

上の管理対象アプリケーションプログラムを停止させるための処理を行う場合に、前記1以上の管理対象アプリケーションプログラムに対して前記コールバック機能の実行を認めることとしてもよく、それによって、不正使用の防止のために管理対象アプリケーションを停止させる場合でも、必要な停止時の後処理は確実に実行される。

【0101】また、本発明のライセンス管理装置は、前記ライセンスキーカーの前記コンピュータシステムからの取り外しを監視し、取り外しを検知した場合は、その旨を前記制御手段に通知するキー監視手段を更に備え、前記制御手段は、前記キー監視手段からの通知を受けると、前記マネージャ手段にライセンスチェック判定処理を実行せることとしてもよく、それによって、ライセンスキーカーの有無を確実に把握できてライセンスチェックがより厳密に行えるとともに、マネージャ手段が定期的にチェックする必要がなくなり、ライセンスチェック処理の負荷が低下し、ライセンスチェックによるシステムのパフォーマンスへの影響は更に小さくできる。

【0102】また、本発明のライセンス管理装置が管理する前記1以上の管理対象アプリケーションプログラムのうち少なくとも1つは制限付き使用が可能であり、前記停止手段は、前記検証手段が制限付き使用の可能な管理対象アプリケーションプログラムについて不正使用と判定した場合、停止処理を行わず、当該制限付き使用が可能な管理対象アプリケーションプログラムについて機能を一部制限するに止めてもよく、それによって、不正使用を防止しつつ、販売促進を目的とした管理対象アプリケーションソフトウェアの試用も可能とすることができる。

【0103】また、本発明のライセンス管理装置は、前記取得手段が前記1以上の管理対象アプリケーションプログラムとの通信によって前記稼動情報を取得し、前記稼動情報は、前記取得手段と前記1以上の管理対象アプリケーションプログラムとのプロセス間通信に用いられる領域であるポートの識別情報を含むこととすることもでき、それによって、マネージャ手段は管理対象アプリケーションソフトウェアを、ポート番号をもとに管理でき、ライセンスチェックのための負荷が軽くなるので、ライセンスチェック処理によるシステムのパフォーマンスの影響を小さくすることができる。

【0104】また、本発明のライセンス管理装置については、前記停止手段が同報通信機能を有し、前記ポートの識別情報を用いて前記1以上の管理対象アプリケーションプログラムに対して終了命令を同報送信することで、前記1以上の管理対象アプリケーションプログラムを停止させることにもよく、それによって、不正使用検知時の管理対象アプリケーションソフトウェアへの停止命令送信の処理を容易に行えるので、ライセンスチェック処理によるシステムのパフォーマンスの影響を小

さくすることができる。

【0105】また、本発明のライセンス管理装置が動作する、前記コンピュータシステムは複数の端末がネットワークで接続されて成り、前記ライセンスキーは、前記複数の端末のうち少なくとも1つに接続され、前記1以上の管理対象アプリケーションプログラムは、前記複数の端末のうち少なくとも1つの上で実行されることとしてもよく、それによって、ネットワーク上でも、管理対象アプリケーションの肥大、さらにはアプリケーションソフトウェアやシステムのパフォーマンス低下を招くことなく、管理対象アプリケーションソフトウェアの不正使用を防止することができる。

【0106】また、本発明のライセンス管理装置は、前記停止手段がシステム権限を有し、前記コンピュータシステム権限を用いて、前記1以上の管理対象アプリケーションプログラムのプロセスを停止させることとしてもよく、それによって、不正使用検知時には、確実に管理対象アプリケーションソフトウェアを停止させることができ可能となり、厳密なライセンス管理を実現できる。

【0107】また、本発明のライセンス管理装置は、前記1以上の管理対象アプリケーションプログラムを子プロセスとして起動および停止させる子プロセス管理手段を更に有し、前記取得手段は、前記子プロセス管理手段から前記稼動情報を取得し、前記停止手段は、前記子プロセス管理手段に指示して前記1以上の管理対象アプリケーションプログラムを停止させることとしてもよく、それによって、確実に管理対象アプリケーションソフトウェアの稼動状態を把握できるとともに、不正使用検知時には、これらを確実に停止させることが可能となるので、厳密なライセンス管理を実現できる。

【0108】そして、やはり以上の説明から明らかなように、本発明のライセンス管理方法は、ライセンス管理の対象となる1以上の管理対象アプリケーションプログラムについてコンピュータシステム上の実行状況を管理するライセンス管理方法であって、前記1以上の管理対象アプリケーションプログラムについて、前記コンピュータシステムでの使用の可否を示すライセンス情報を保持する、ハードウェアのライセンスキーから前記ライセンス情報を読み出す読み出しステップと、前記1以上の管理対象アプリケーションプログラムの前記コンピュータシステム上の実行状況を示す稼動情報を取得する取得ステップと、前記読み出しステップにおいて読み出されたライセンス情報と前記取得ステップにおいて取得された稼動情報をもとに、前記1以上の管理対象アプリケーションプログラムの中に、前記コンピュータシステムでの動作が認可されていないにもかかわらず動作しているものがあれば、不正使用ありと判定する判定ステップと、前記判定ステップにおいてが不正使用ありと判定された場合、稼働中である前記1以上の管理対象アプリケーションプログラムのうち、少なくとも前記判定ス

テップにおいて前記コンピュータシステムでの動作が認可されていないにもかかわらず動作していると判定されたものを停止させるための処理を行う停止ステップとを有することを特徴とする。よって、管理対象アプリケーションが別個に内蔵されたトリガ命令を実行する形でライセンスチェックを行うことがないので、管理対象アプリケーションの肥大、さらにはアプリケーションソフトウェアやシステムのパフォーマンス低下を招くことなく、管理対象アプリケーションソフトウェアの不正使用を防止できる。

#### 【図面の簡単な説明】

【図1】本発明が適用されたライセンス管理装置の実施の形態において、本ライセンス管理装置がPC上でどのように実現されるかを示す模式図である。

【図2】同実施の形態においてライセンスキーが保持するライセンス情報の一例を示す図である。

【図3】同実施の形態におけるライセンスマネージャの構成を示すブロック図である。

【図4】同実施の形態における使用可否情報の例を示す。

【図5】同実施の形態における稼動情報の構成および内容の一例を示す。

【図6】同実施の形態におけるライセンスマネージャの動作を示すフローチャートである。

【図7】同実施の形態の第1の変形例におけるライセンスマネージャの動作を示すフローチャートである。

【図8】同実施の形態の第2変形例において、不正使用検知にともなう管理対象アプリケーション停止時に、ライセンスマネージャ、管理対象アプリケーション、ライセンスキーの間で交わされるメッセージシーケンスを示す図である。

【図9】同実施の形態の第3変形例におけるライセンスマネージャと管理対象アプリケーションとの関係を示す階層図である。

【図10】同変形例におけるライセンスマネージャの構成を示すブロック図である。

【図11】同実施の形態の第5変形例におけるライセンス管理装置をネットワーク上の複数のPC上で動作させるネットワークシステムを示す図である。

【図12】同変形例においてライセンスキーが保持するライセンス情報の内容を示す図である。

【図13】同変形例における稼動情報の構成および内容の一例を示す図である。

【図14】同変形例におけるライセンスマネージャの動作を示すフローチャートである。

【図15】同実施の形態の第6変形例におけるライセンス管理装置の構成を示す図である。

【図16】同変形例における停止命令発行部が出力させる強制停止の通知メッセージの一例を示す図である。

【図17】本変形例におけるライセンスマネージャの動

作を示すフローチャートである。

【図18】従来技術であるインストール時のライセンスチェックの概要を示す模式図である。

【図19】従来技術であるアプリケーション起動時のライセンスチェックの概要を示す模式図である。

【図20】従来技術として、管理対象アプリケーションに埋め込まれたトリガ命令を示すイメージ図である。

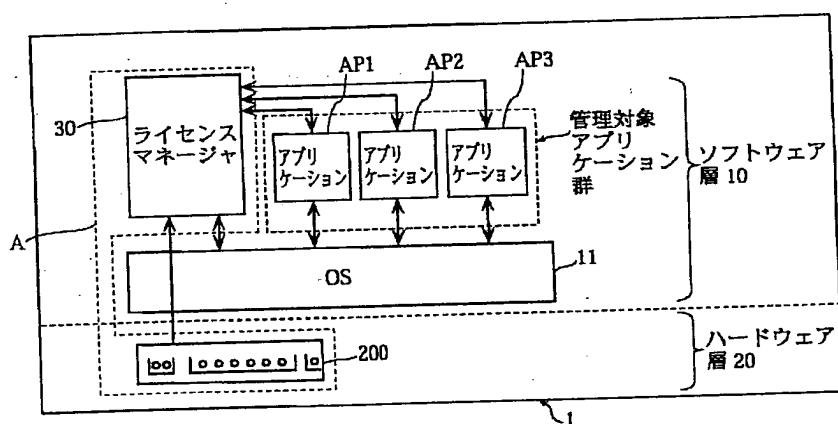
【図21】従来技術として、トリガ命令を埋め込まれた管理対象アプリケーションが1つのPC上で同時に稼動している状態を示す模式図である。

#### 【符号の説明】

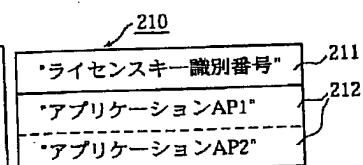
A, B ライセンス管理装置

1 PC

【図1】

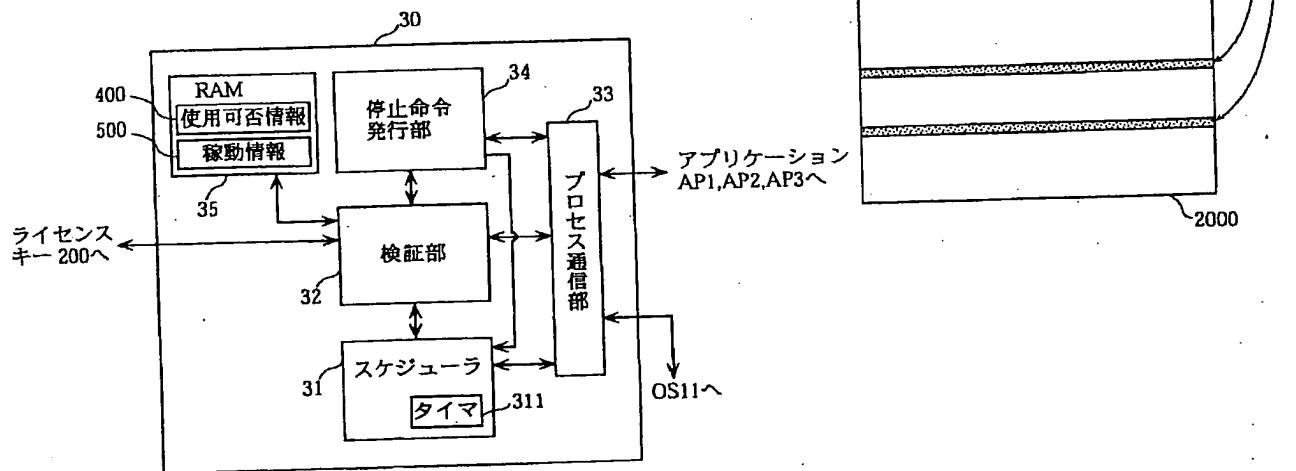


【図2】

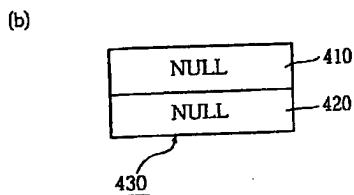
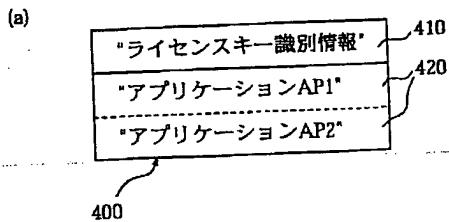


【図20】

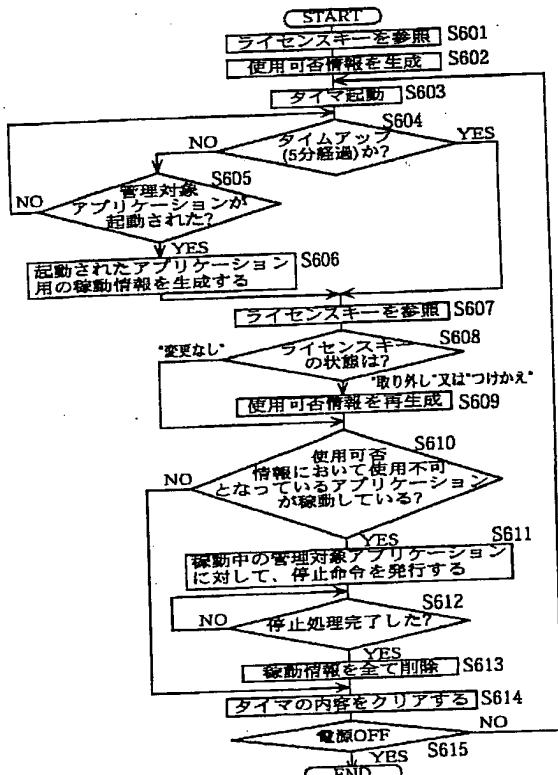
【図3】



【図4】



【図6】

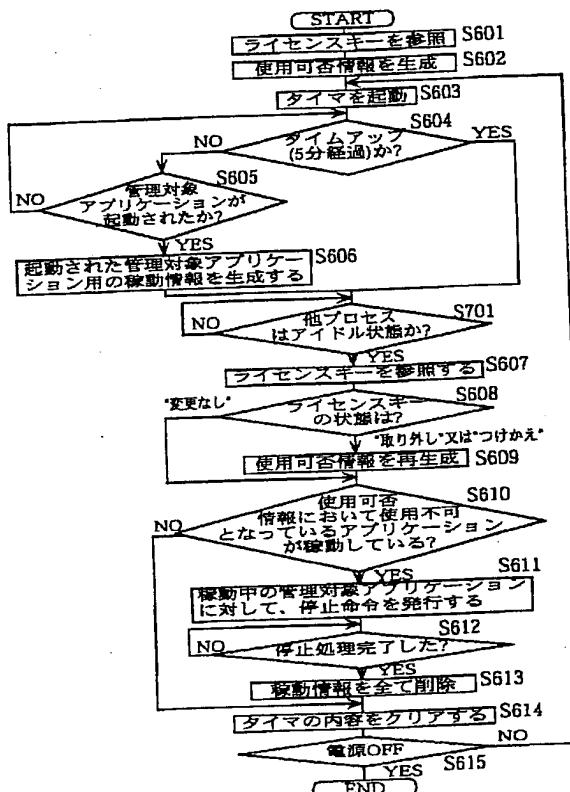


【図5】

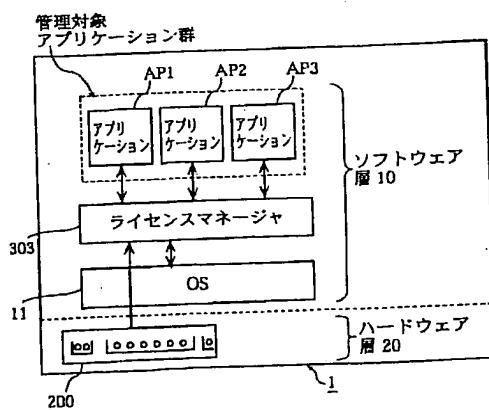
アプリケーションID欄	ポートNo.欄	プロセスID欄	500
アプリケーションAP1	2001	101	
アプリケーションAP3	2002	103	

510 520 530

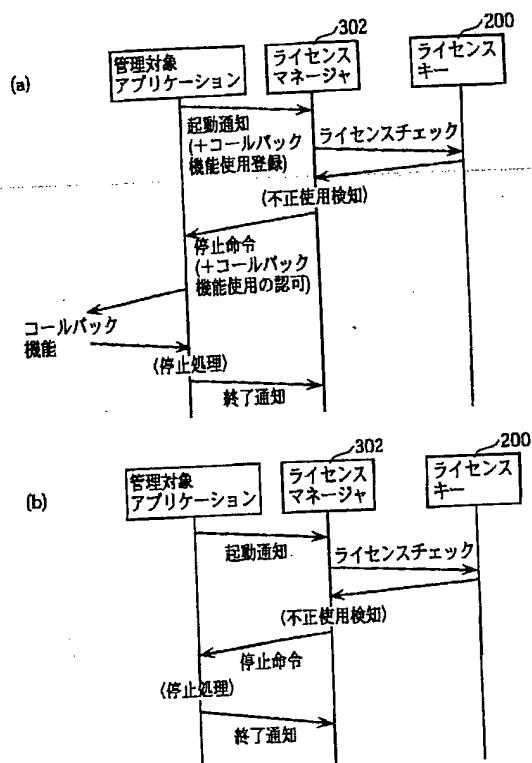
【図7】



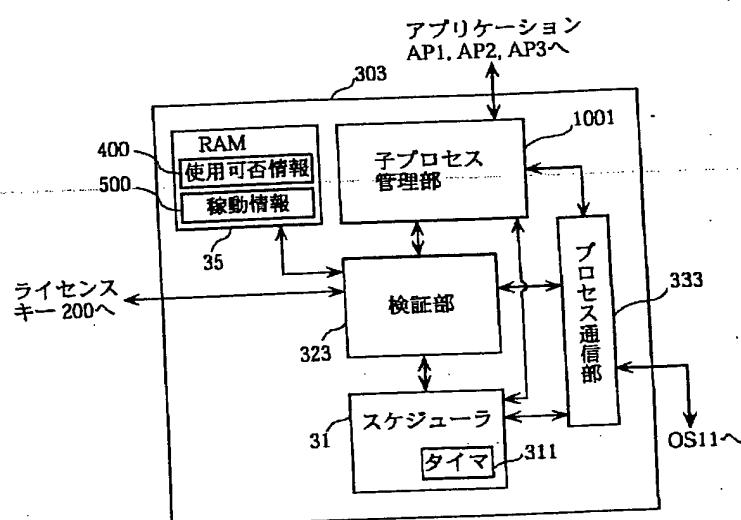
【図9】



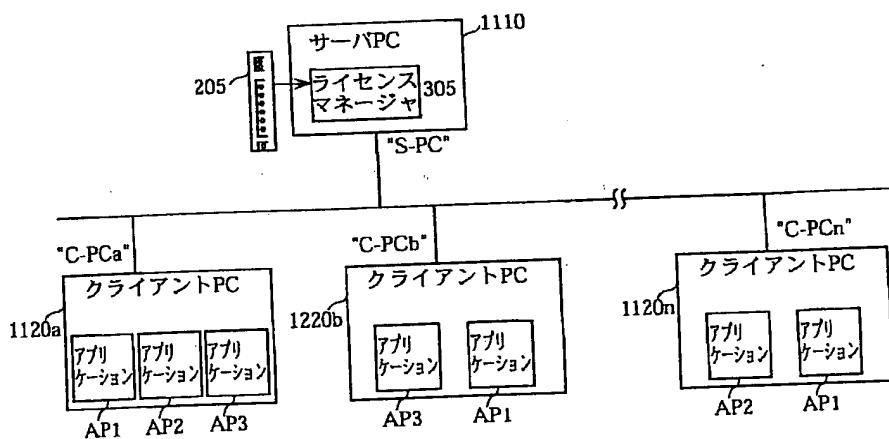
【図 8】



【図 10】



【図 11】



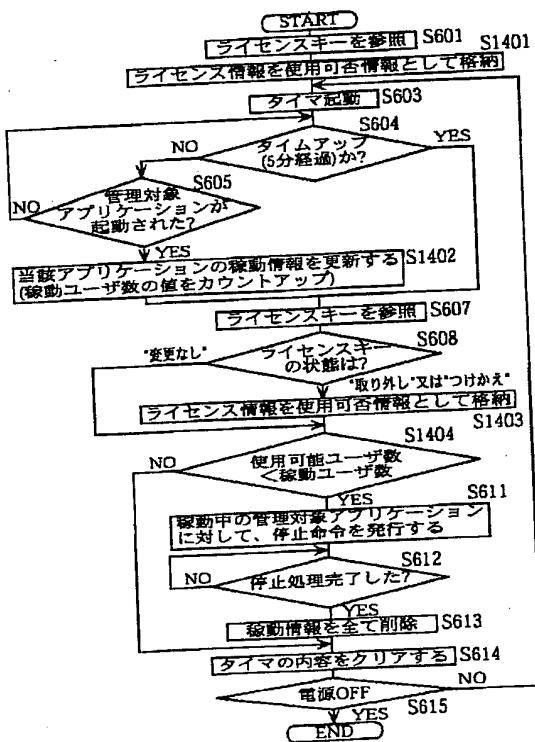
【図 12】

1200		1210
1220a→	・ライセンスキード識別情報	1222a
1220b→	・アプリケーションAP1*	5
1220b→	・アプリケーションAP2*	4
1220c→	・アプリケーションAP3*	1
		1222b 1222c
		1221a 1221b 1221c

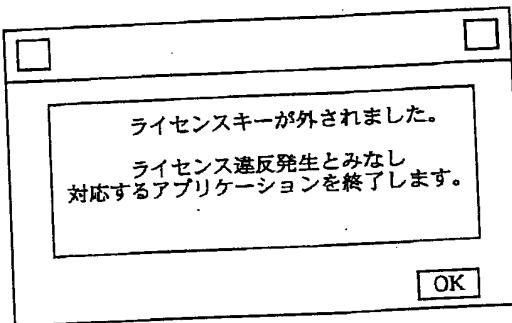
【図13】

アプリケーションID欄	稼動ユーザ数欄	ポートNo.欄	プロセスID欄
"アプリケーション AP1"	3	"C-PCa""2001" "C-PCb""2002" "C-PCr""2002"	"C-PCa""101" "C-PCb""102" "C-PCr""102"
"アプリケーション AP2"	2	"C-PCa""2002" "C-PCn""2001"	"C-PCa""102" "C-PCn""101"
"アプリケーション AP3"	2	"C-PCa""2003" "C-PCb""2001"	"C-PCa""103" "C-PCb""101"

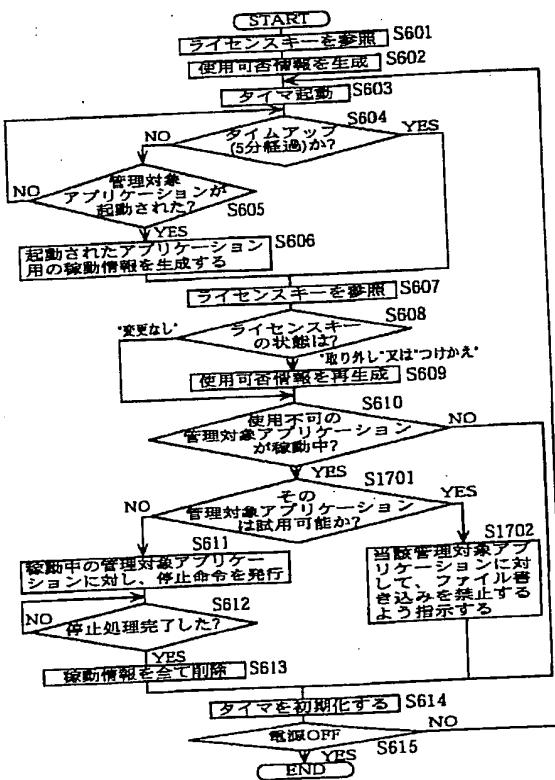
【図14】



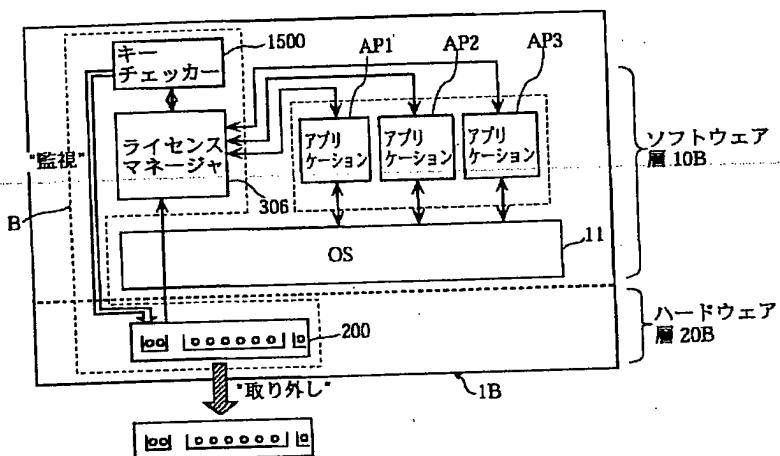
【図16】



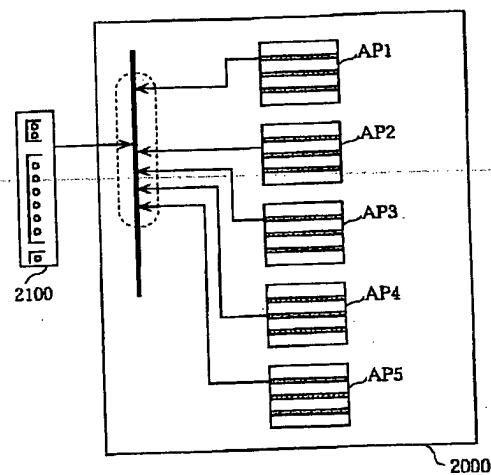
【図17】



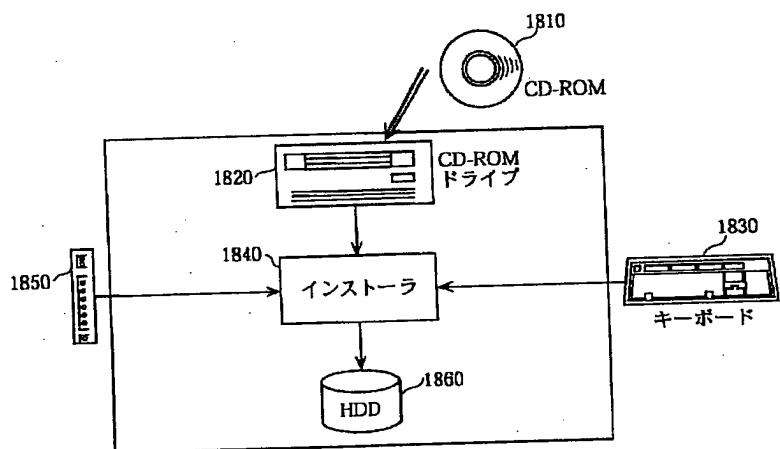
【図15】



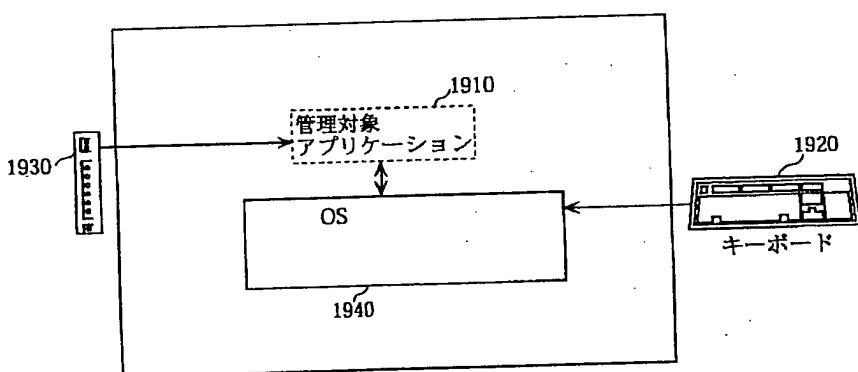
【図21】



【図18】



【図19】



## フロントページの続き

(72) 発明者 鈴木 基史

愛知県名古屋市中区栄2丁目6番1号 白

川ビル別館5階 株式会社松下電器情報シ

ステム名古屋研究所内

Fターム(参考) 5B076 FB07